Scientific Computing, Modeling & Simulation
Savitribai Phule Pune University.

Machine Learning Mini Project.

Topic - Hybrid Movie Recommender System Combining Content-Based and Collaborative Filtering Methods.

Name – Sarang Pekhale.

Roll No. – MT2212.

Under the Guidance of – *Prof. Dr. Mihir Arjunwadkar.*

Introduction

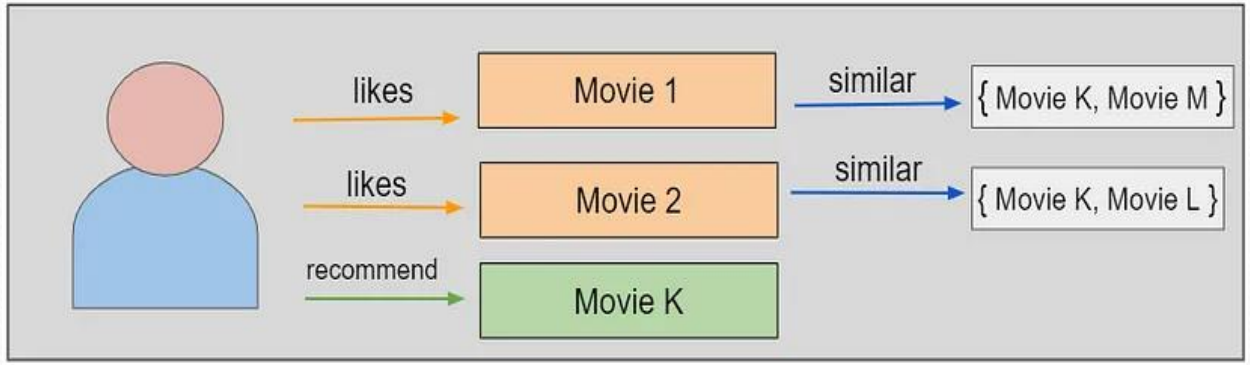
1. Overview of Movie Recommender Systems :
   - What : To recommend relevant and engaging content.
   - Why : For customer entertainment, satisfaction and retention.
   - How : A culmination of content-based and collaborative filtering methods.
2. Objective :
   - To provide movie enthusiasts with recommendations that are not only tailored to their individual preferences but also diverse and engaging.

Problem Statement

- The challenge is to develop a hybrid movie recommender system that combines content-based and collaborative filtering methods to address the following key issues:
  1. Information Overload.
  2. Diverse User Preferences.
  3. Cold Start Problem.
  4. Limited Exploration.

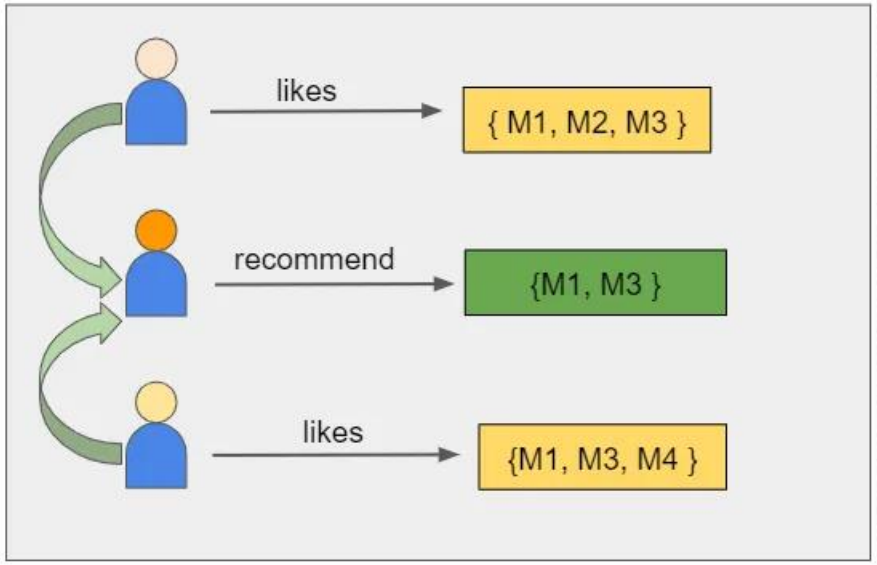
Content-Based Method

- Content-based movie recommender systems operate on the principle of analyzing the content features associated with each movie in the catalog. The process can be broken down into the following steps:
  1. Feature Extraction.
  2. User Profile Creation.
  3. Content Matching.
  4. Recommendation Generation.



Collaborative Filtering Method

- Collaborative filtering-based movie recommender systems operate by understanding the underlying patterns in user interactions. The process can be summarized as follows:
  1. User-Item Interaction Matrix.
  2. User-Based Collaborative Filtering.
  3. Item-Based Collaborative Filtering.
  4. Recommendation Generation.



Data : TMDB 5000 Movie Dataset

- Description :
  1. Movie Metadata.
  2. Crew and Cast Information.
  3. Keywords.
  4. User Ratings and Popularity Metrics.
  5. Production Companies.
  6. Runtime and Budget Information.
  7. Overview and Tagline.
  8. Spoken Languages and Release Regions.
- Relevance :
  1. Rich Movie Feature Data.
  2. User Ratings and Popularity:.
  3. Cast and Crew Details.
  4. Diverse Movie Selection.
  5. Real-World Relevance.
  6. Multilingual and Global Context.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4803 entries, 0 to 4802
Data columns (total 22 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   id                    4803 non-null   int64  
 1   cast                  4803 non-null   object 
 2   crew                  4803 non-null   object 
 3   budget                4803 non-null   int64  
 4   genres                4803 non-null   object 
 5   homepage              1712 non-null   object 
 6   keywords              4803 non-null   object 
 7   original_language     4803 non-null   object 
 8   original_title        4803 non-null   object 
 9   overview              4800 non-null   object 
 10  popularity            4803 non-null   float64
 11  production_companies  4803 non-null   object 
 12  production_countries  4803 non-null   object 
 13  release_date          4802 non-null   object 
 14  revenue               4803 non-null   int64  
 15  runtime               4801 non-null   float64
 16  spoken_languages      4803 non-null   object 
 17  status                4803 non-null   object 
 18  tagline               3959 non-null   object 
 19  title                 4803 non-null   object 
 20  vote_average          4803 non-null   float64
 21  vote_count            4803 non-null   int64  
dtypes: float64(3), int64(4), object(15)
memory usage: 825.6+ KB
```

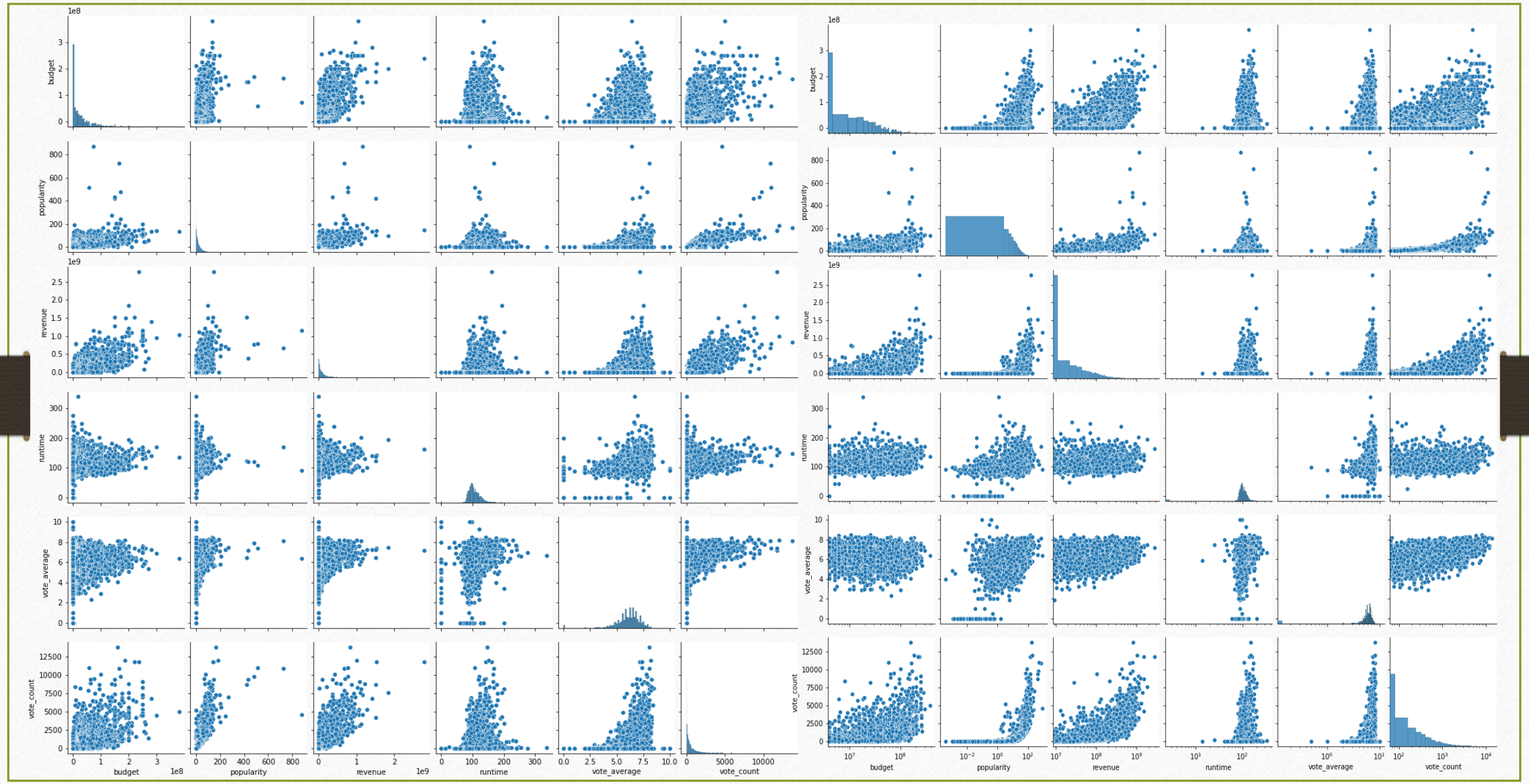
Data Preprocessing for EDA

1. ’df-credits’ (4803 rows, 4 columns) and ’df-movies’ (4803 rows, 20 columns).
2. Renamed the ’movie-id’ column to ’id’ for consistency.
3. Removed duplicate columns : ’title’ column from the CREDITS dataset.
4. Merge : CREDITS and MOVIES, using the common ’id’ column. ’df,’ (4803 rows, 22 columns).
5. Eliminated ’homepage,’ column having more missing values than usual.
6. Recognizing that ’original-title’ and ’title’ are nearly identical, we opted to remove the ’original-title’ column.
7. Extract the textual data from the dictionaries in the observations of these columns. ”keywords,” ”genres,” ”production-companies,” and ”production-countries.”
8. ”release-date” column to ”yyyy/mm/dd”. ”year,” ”month,” and ”day,” from the ”release-date”.
9. Integrated the movie titles with their release dates.
10. Convert the JSON-formatted ”cast” and ”crew” data.

Exploratory Data Analysis

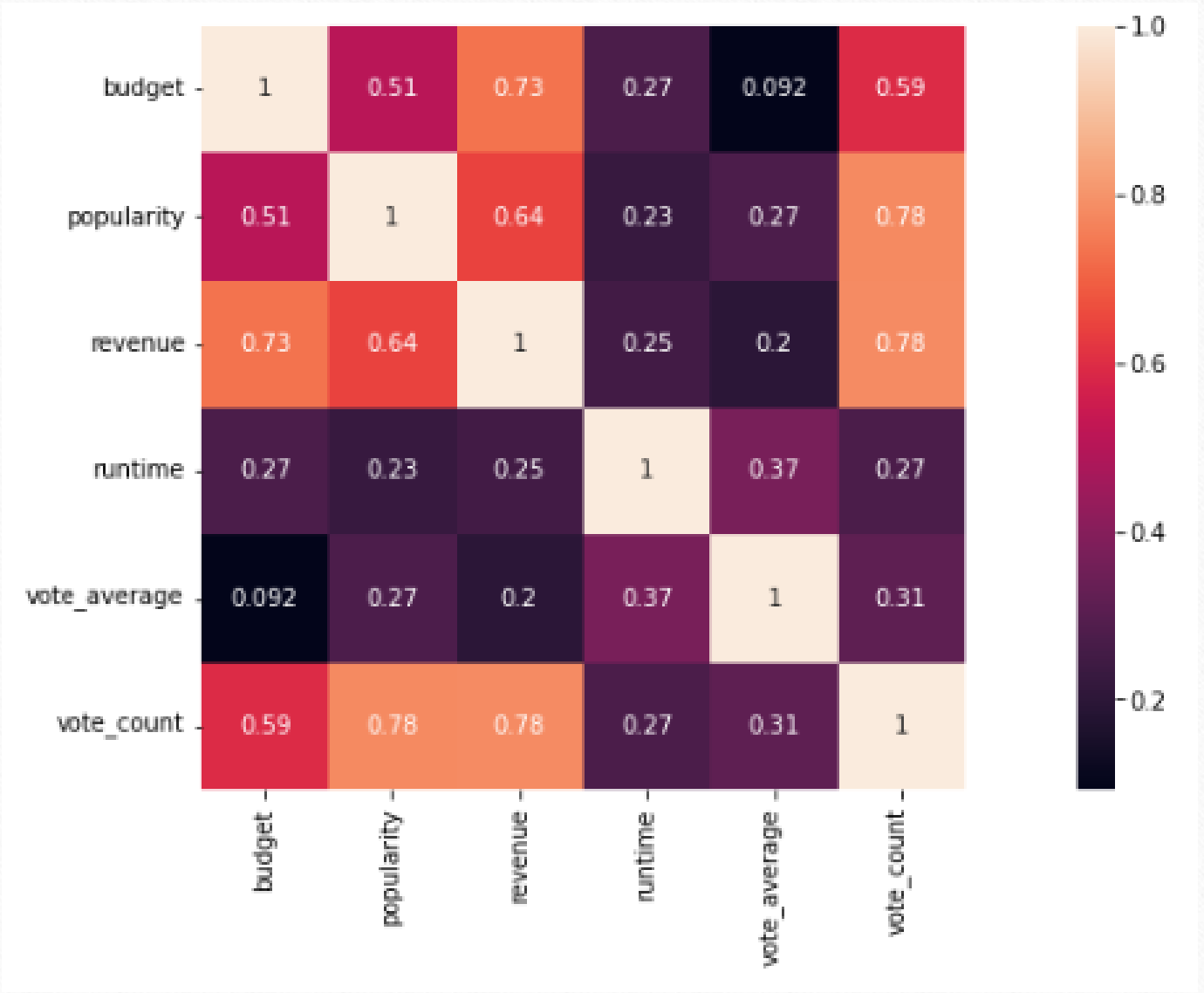
- Descriptive statistics of Data :

	id	budget	popularity	revenue	runtime	vote_average	vote_count
count	4803.000000	4.803000e+03	4803.000000	4.803000e+03	4801.000000	4803.000000	4803.000000
mean	57165.484281	2.904504e+07	21.492301	8.226064e+07	106.875859	6.092172	690.217989
std	88694.614033	4.072239e+07	31.816650	1.628571e+08	22.611935	1.194612	1234.585891
min	5.000000	0.000000e+00	0.000000	0.000000e+00	0.000000	0.000000	0.000000
25%	9014.500000	7.900000e+05	4.668070	0.000000e+00	94.000000	5.600000	54.000000
50%	14629.000000	1.500000e+07	12.921594	1.917000e+07	103.000000	6.200000	235.000000
75%	58610.500000	4.000000e+07	28.313505	9.291719e+07	118.000000	6.800000	737.000000
max	459488.000000	3.800000e+08	875.581305	2.787965e+09	338.000000	10.000000	13752.000000



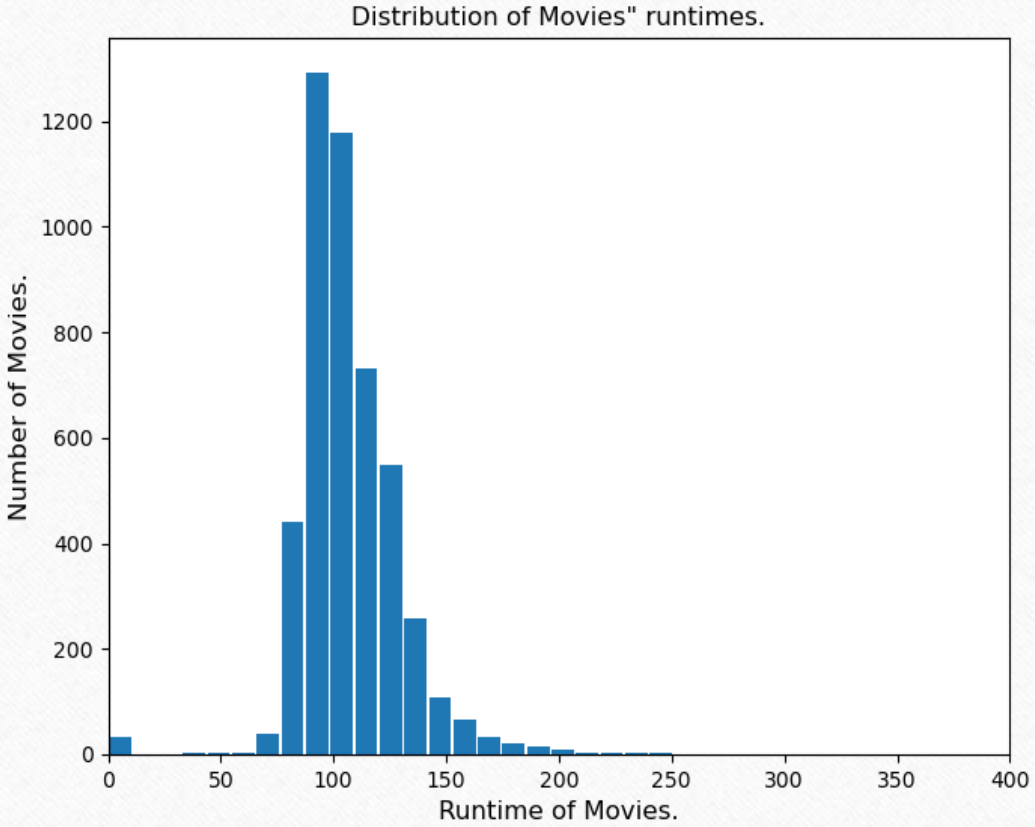
❖ Pairplot of Data.

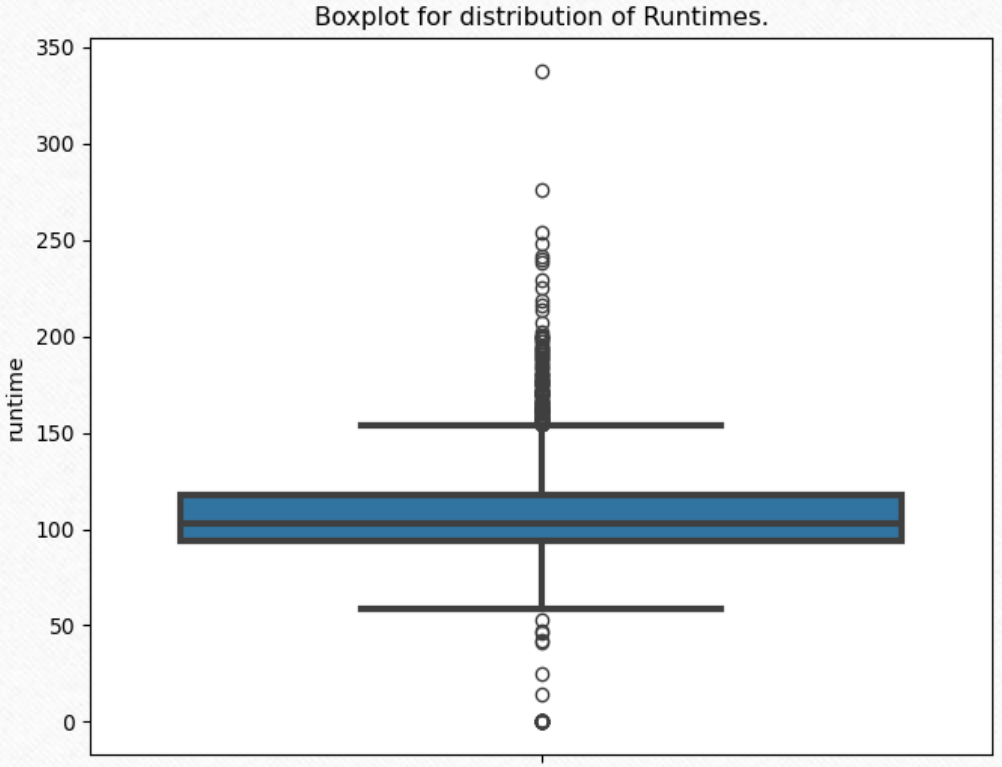
❖ Pairplot of Log scaled Data.



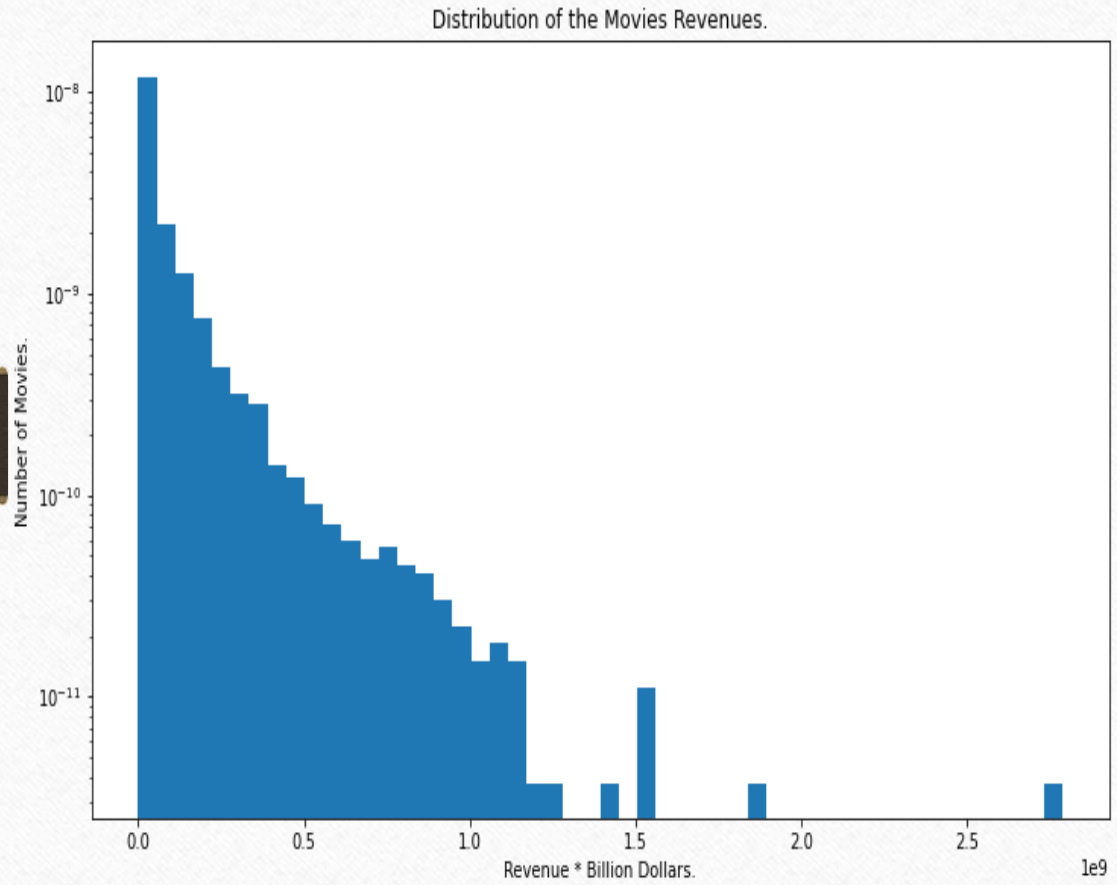
❖ Correlation Matrix of Data.

❖ Movies Runtime.

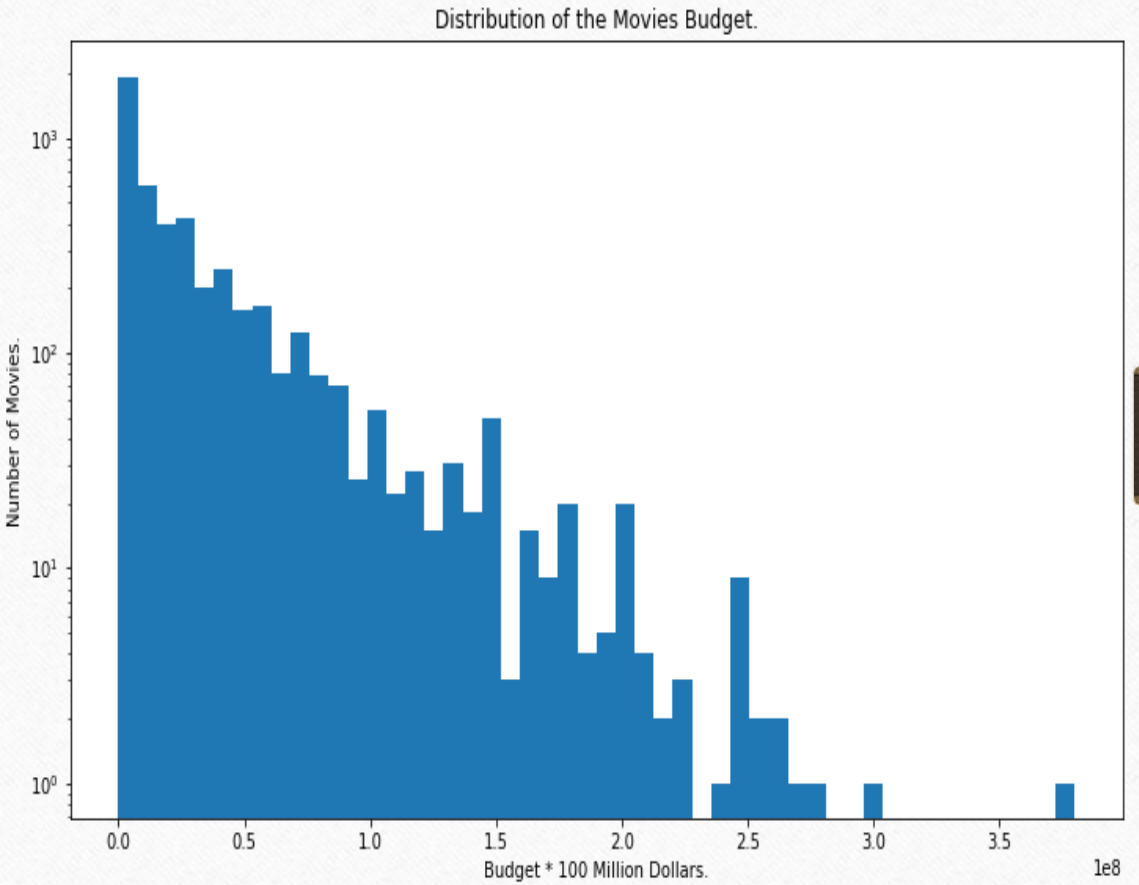




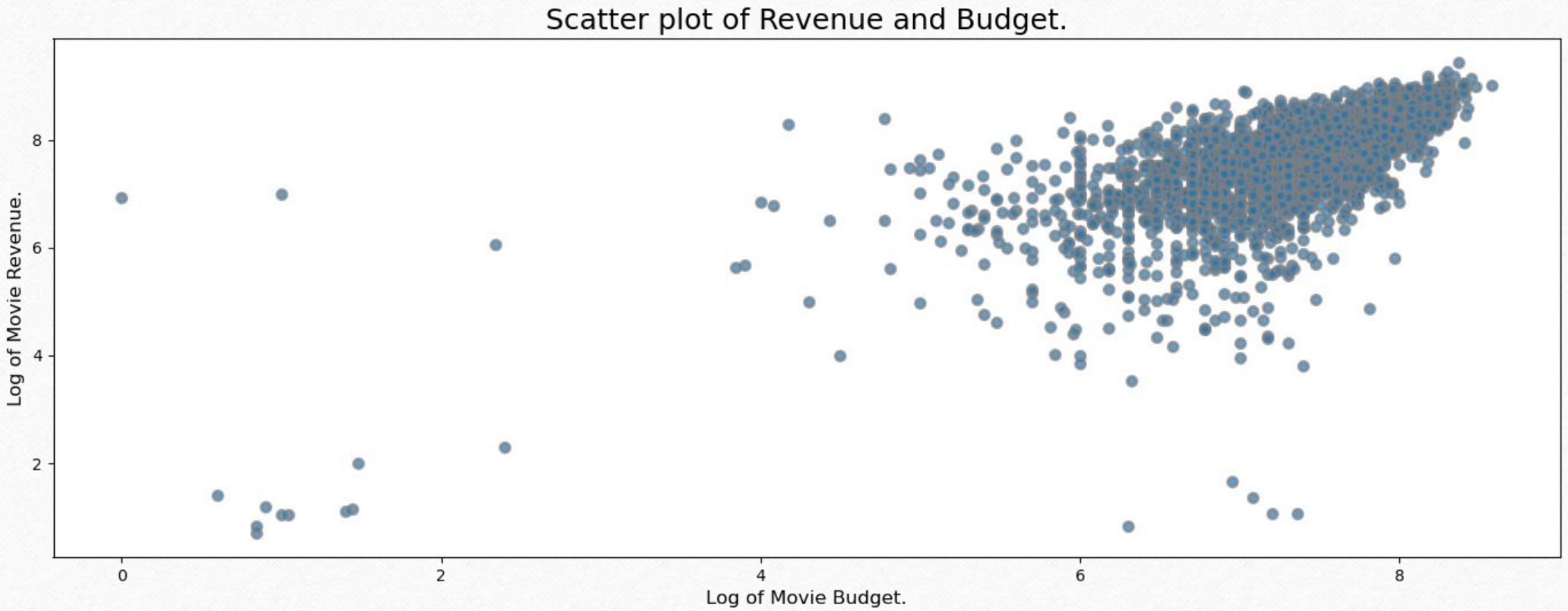
❖ Movies Revenues Distribution.



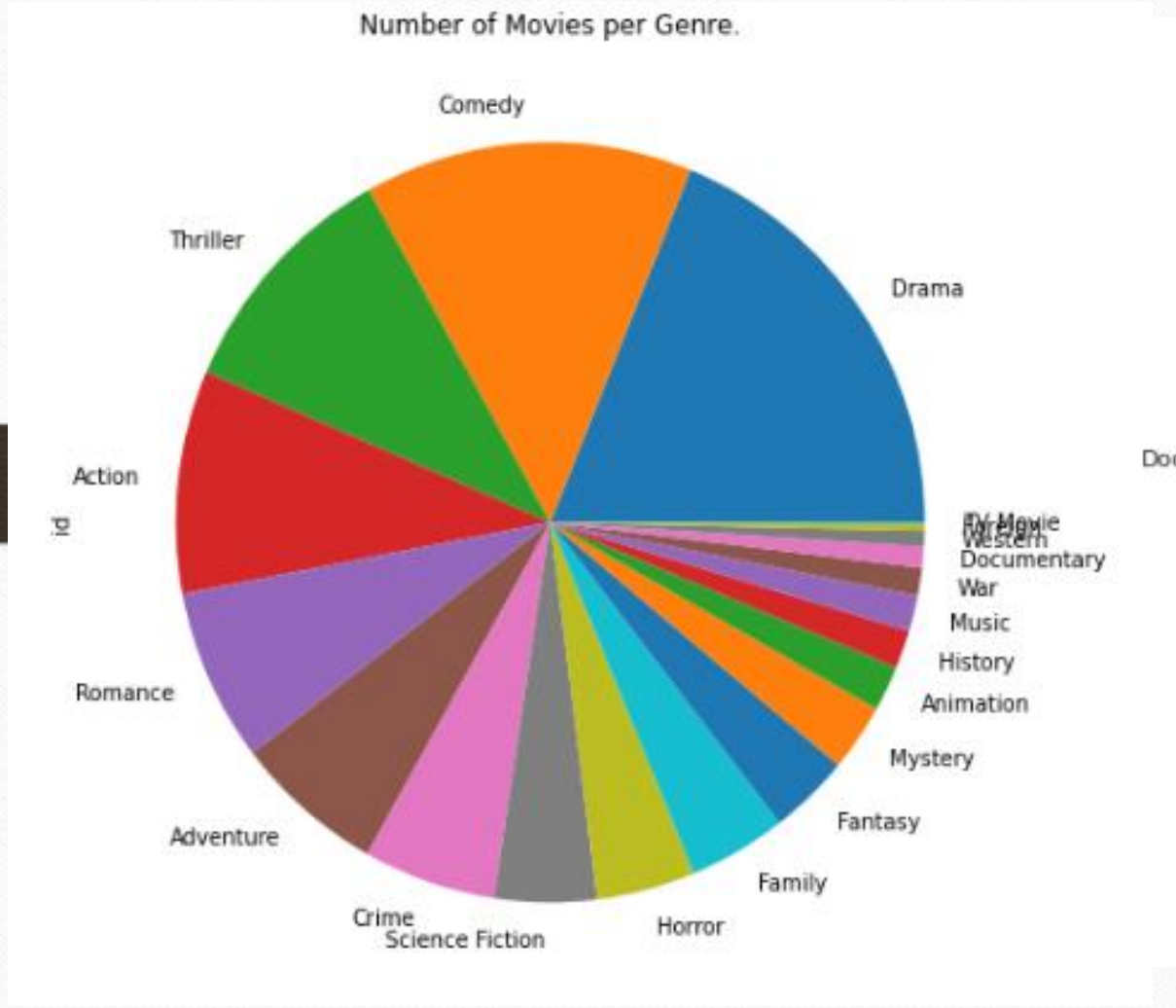
❖ Movies Budget Distribution.



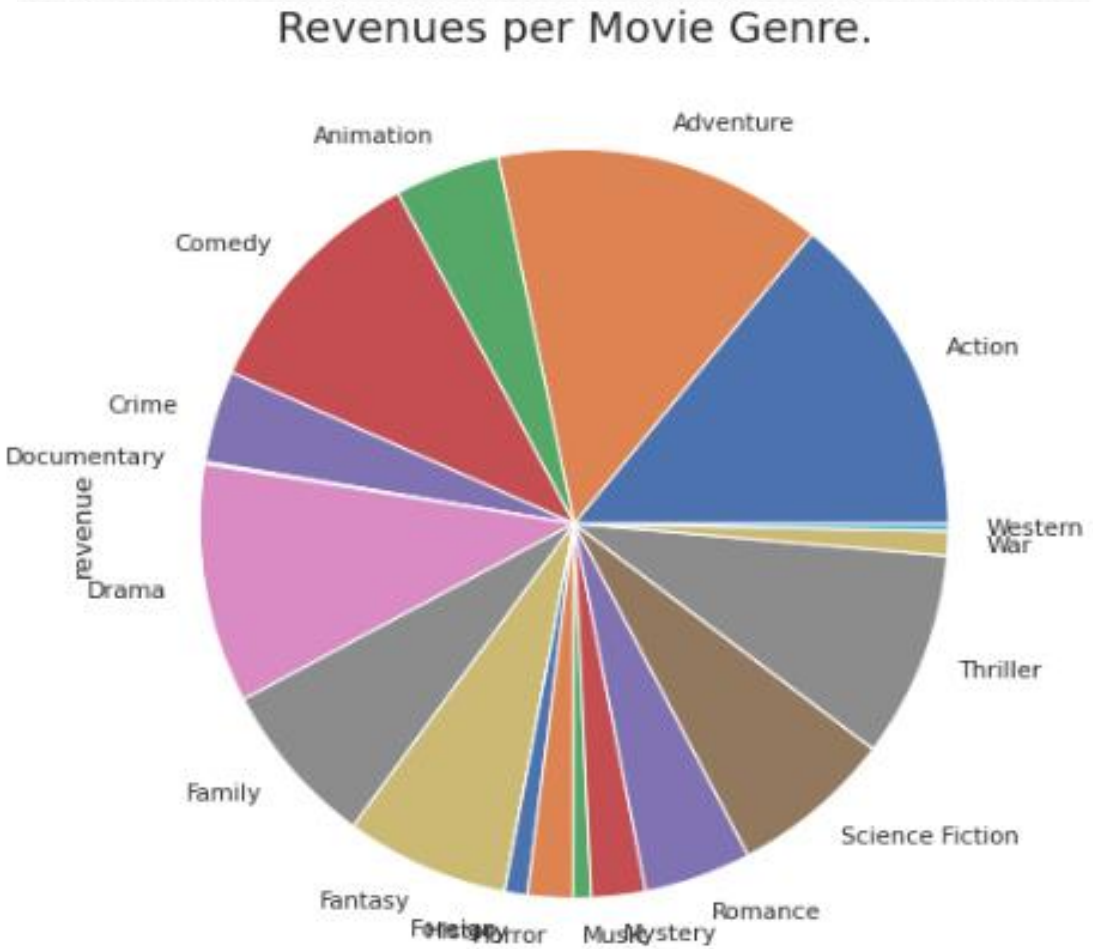
❖ Budget vs Revenue.



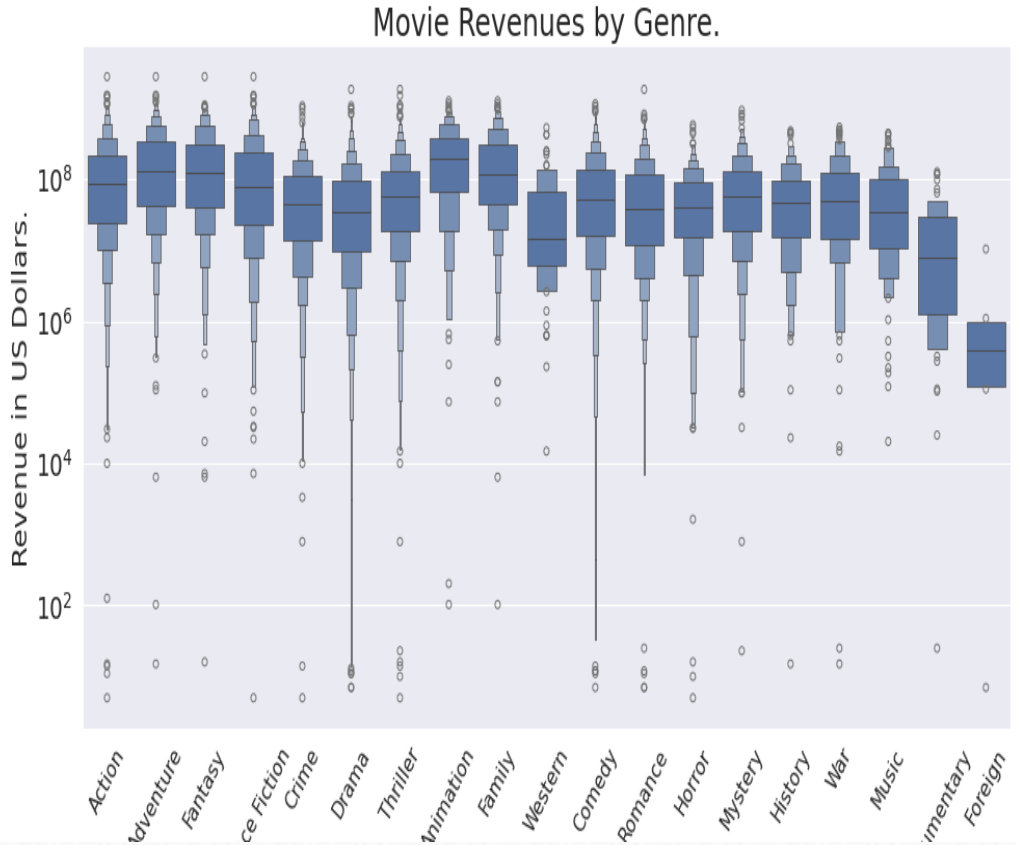
❖ Movies by Genres.



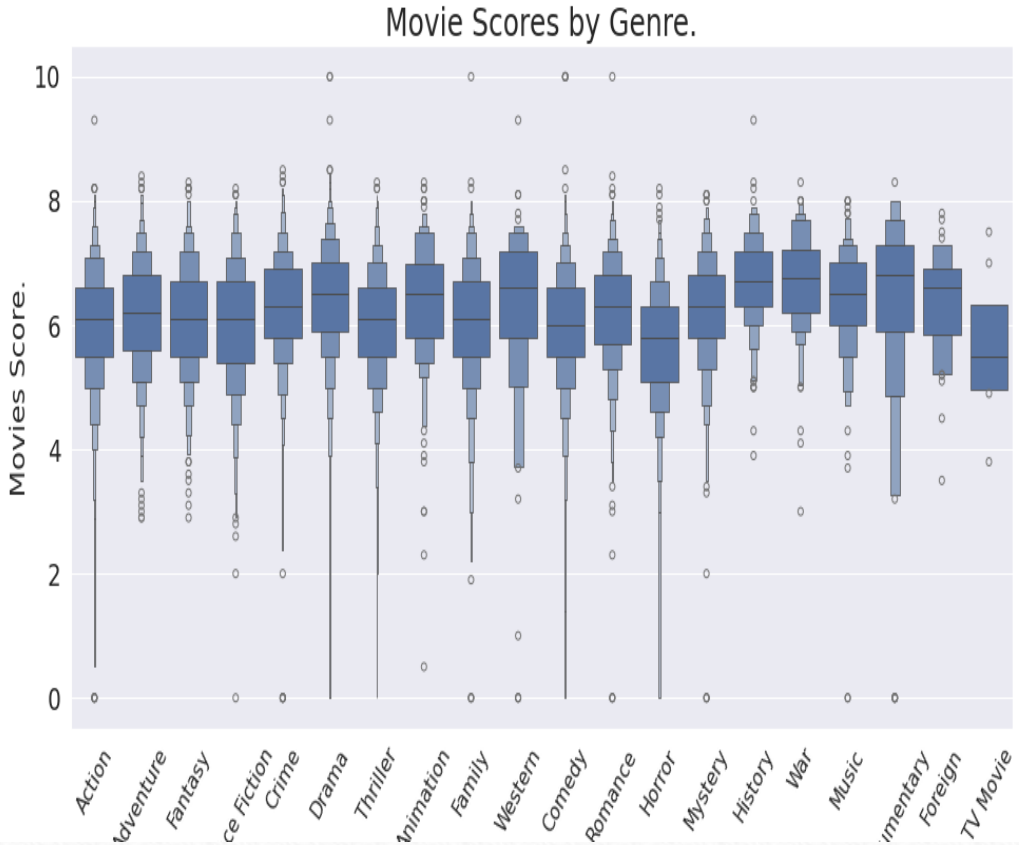
❖ Movie Revenue by Genres.



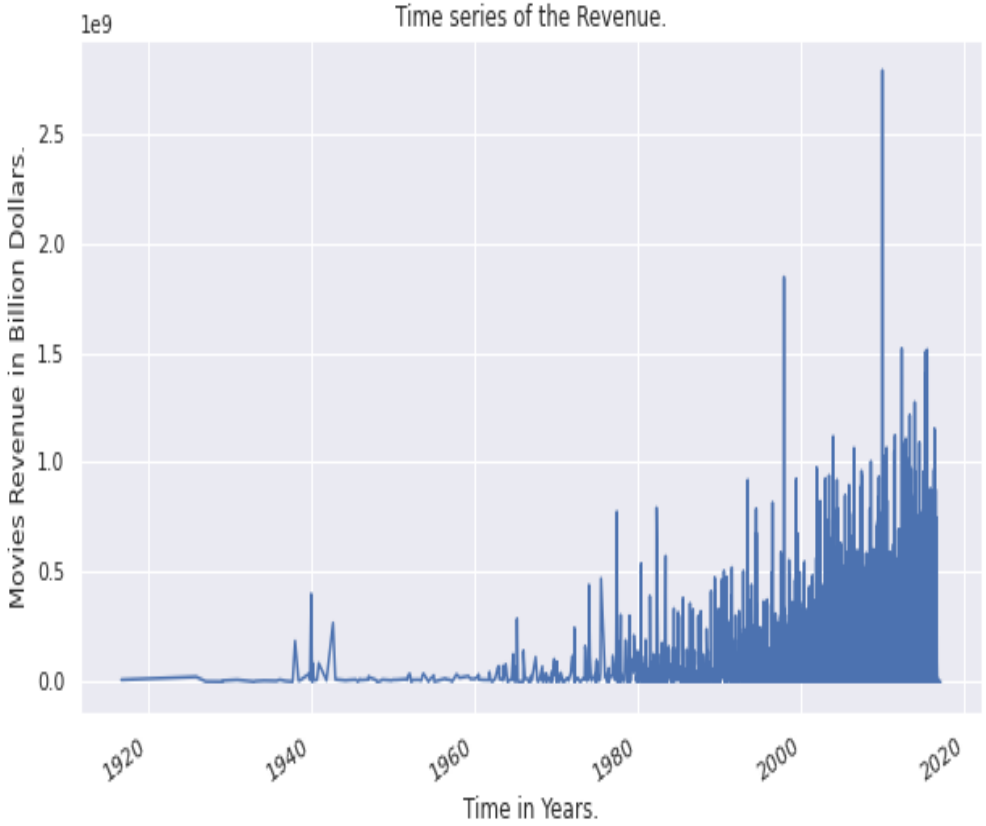
❖ Movies Revenues by Generes.



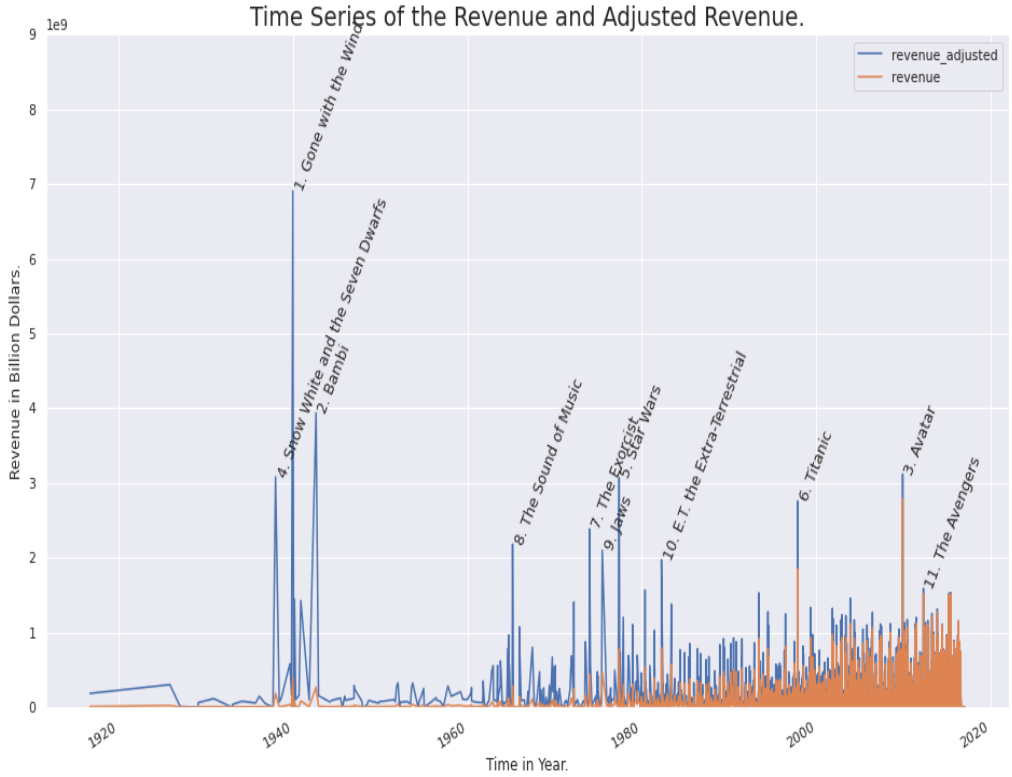
❖ Movies Scores by Genres.



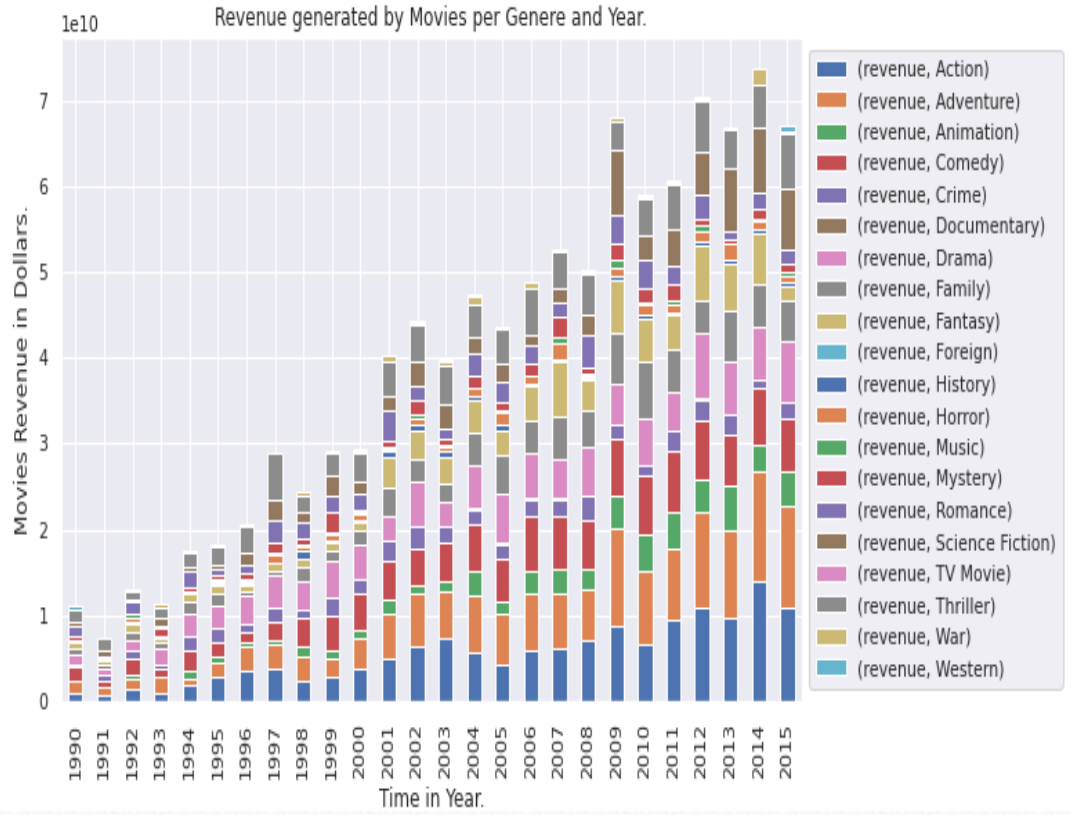
❖ Movies Revenue vs Release Year.



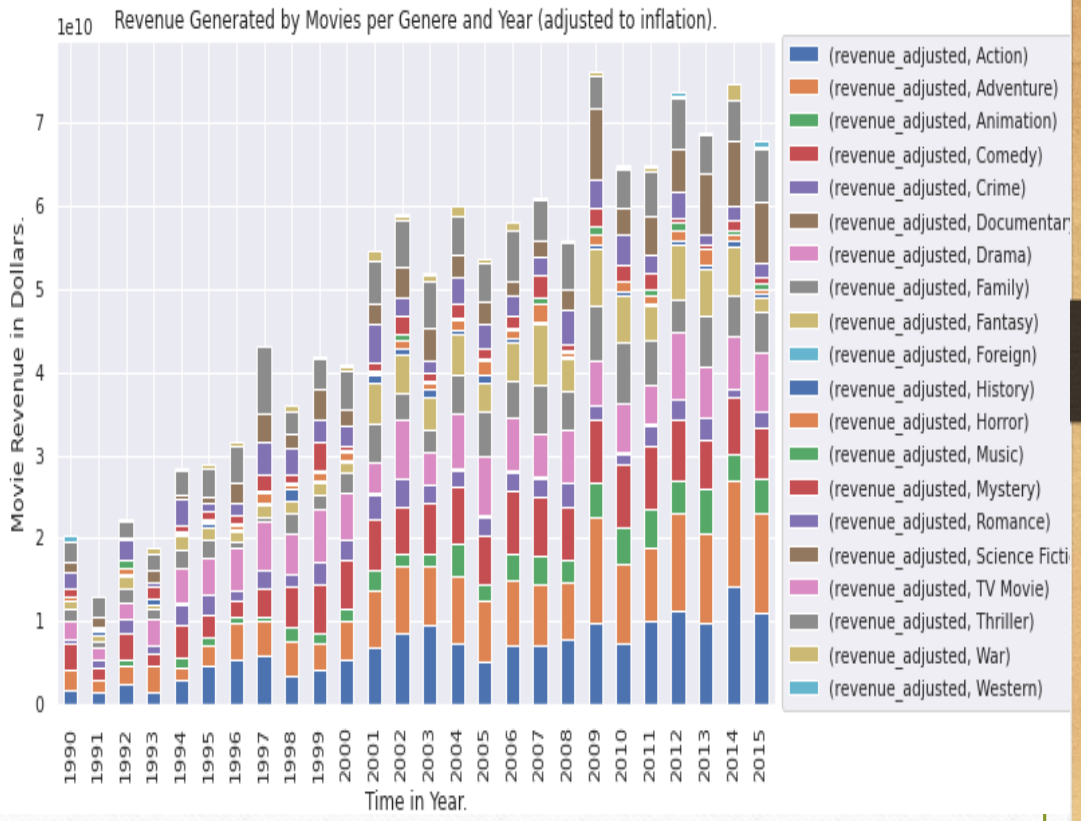
❖ Movies Revenue and Adjusted Revenue vs Movie Year.



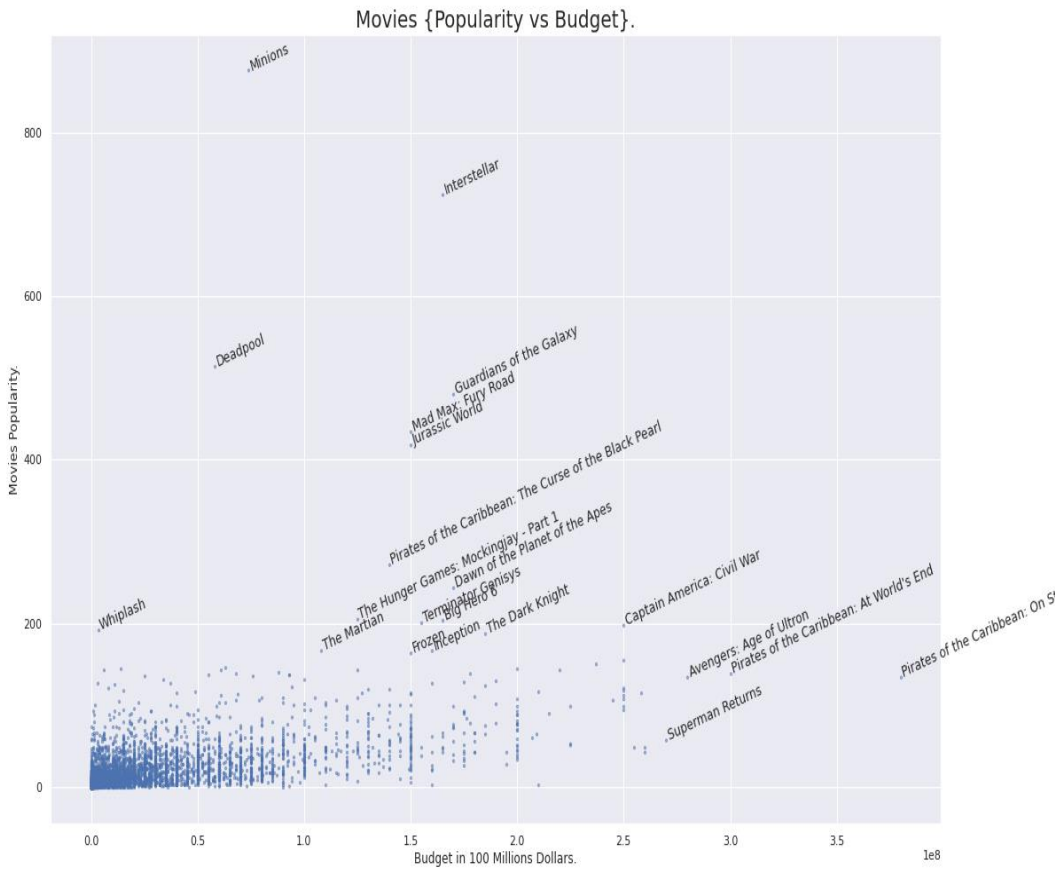
❖ Movies Revenue and Release Year with Genres.



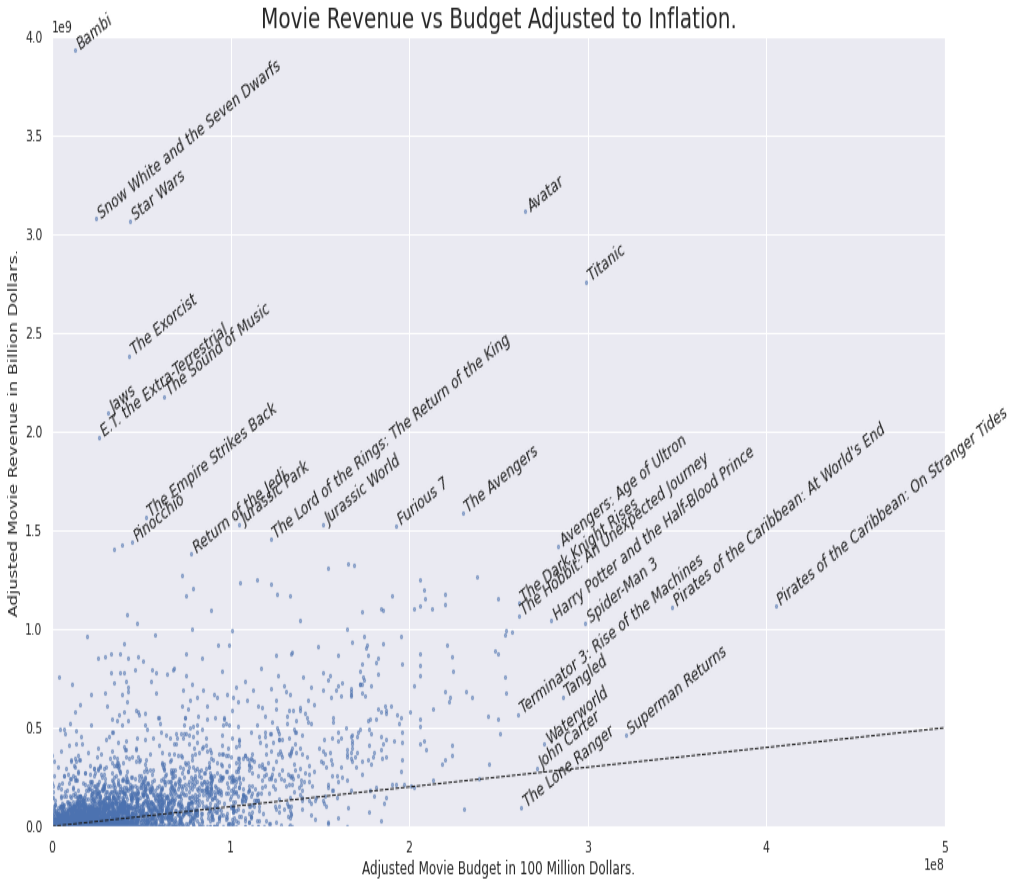
❖ Adjusted Movies Revenue and Release Year with Genres.



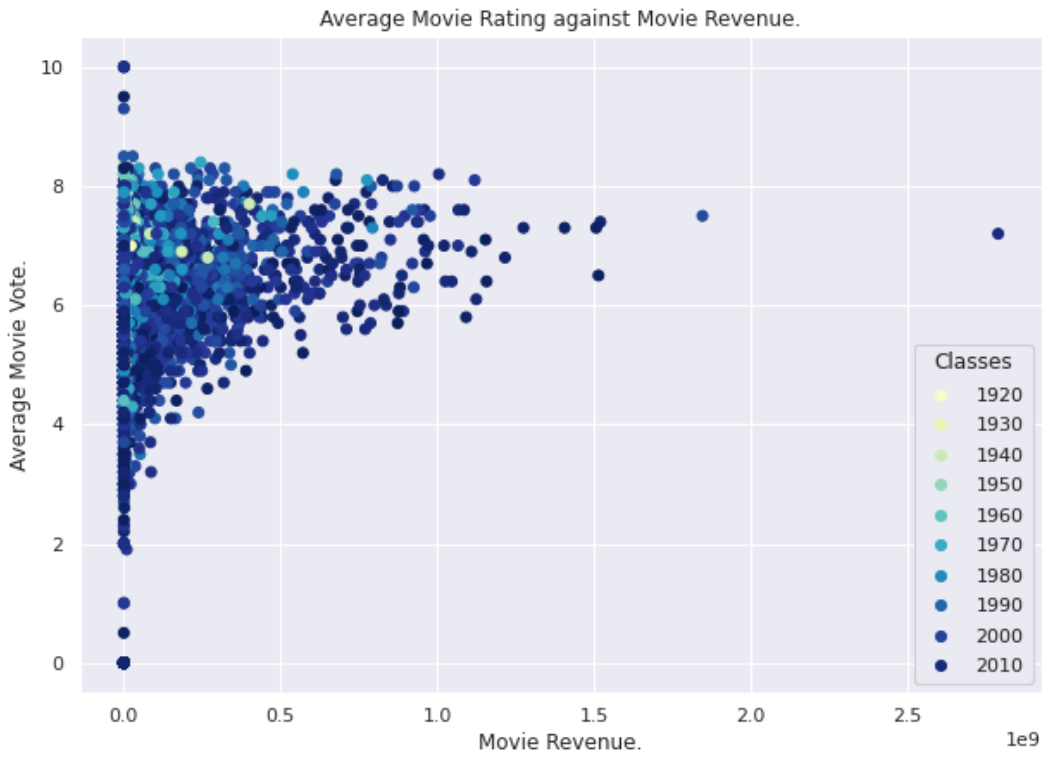
❖ Movies Popularity vs Budget.



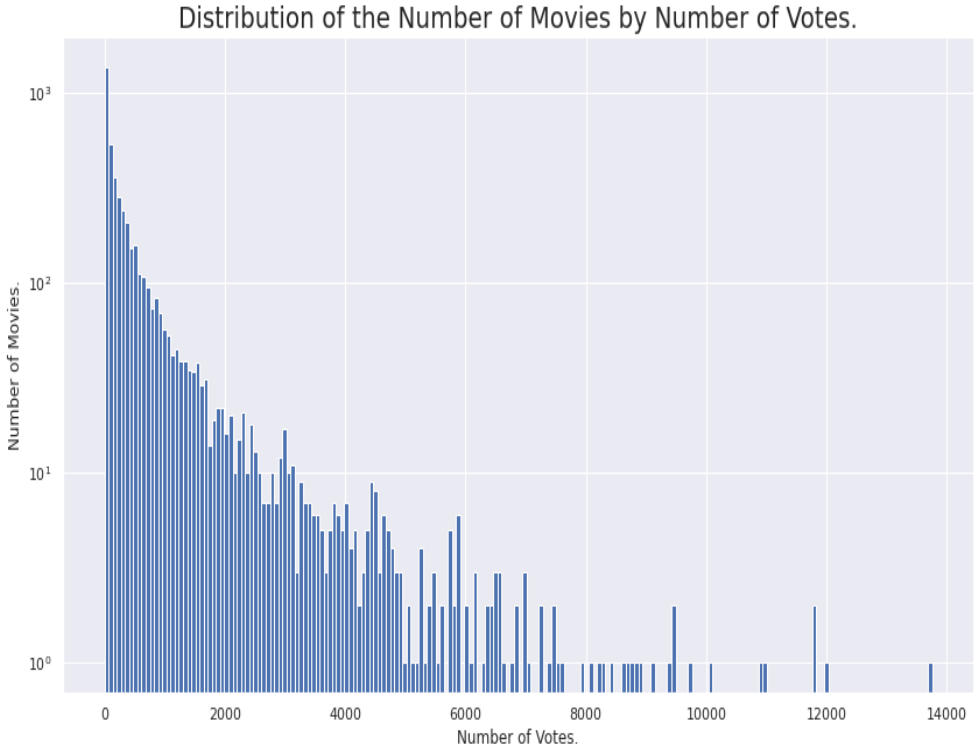
❖ Movie Revenue vs Movie Budget.



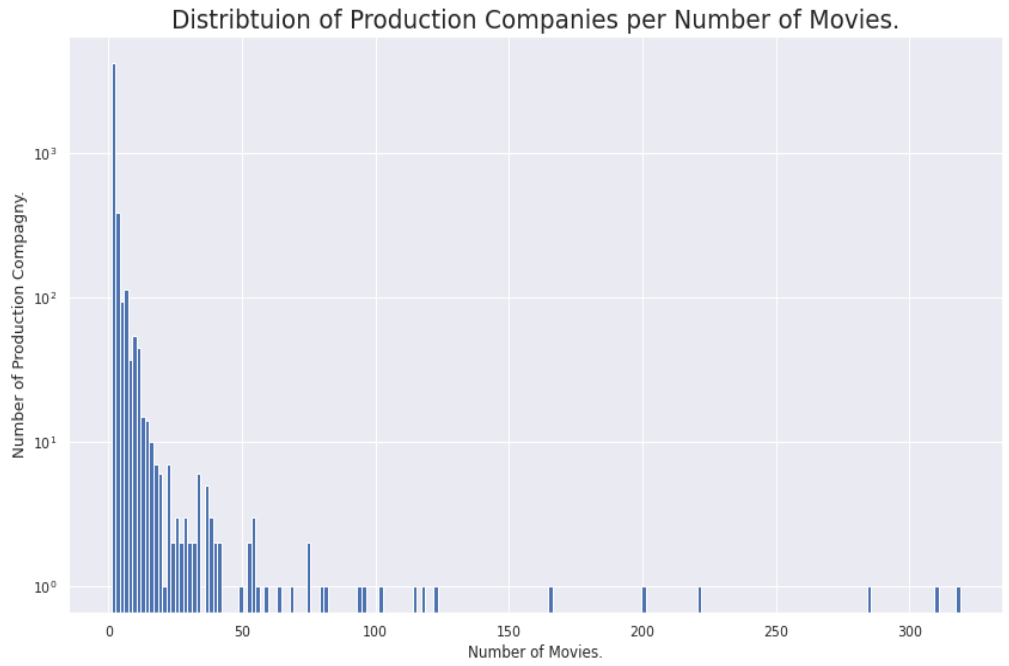
❖ Average Movie Rating vs Movie Revenue.



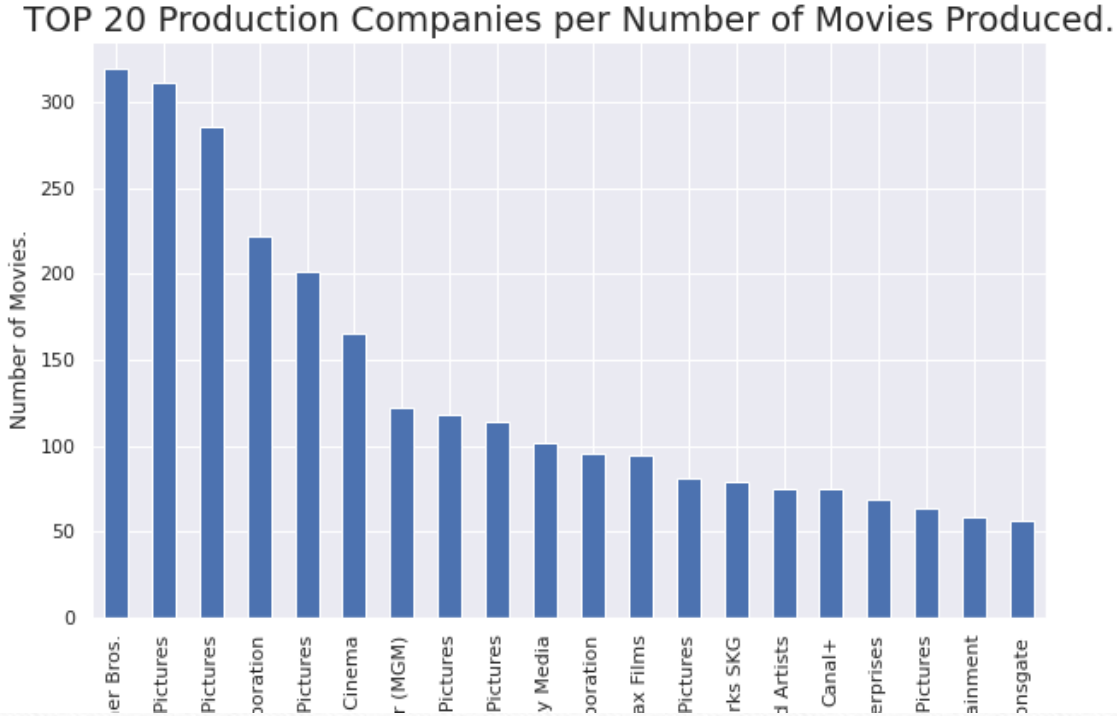
❖ Movie Rating Distribution.



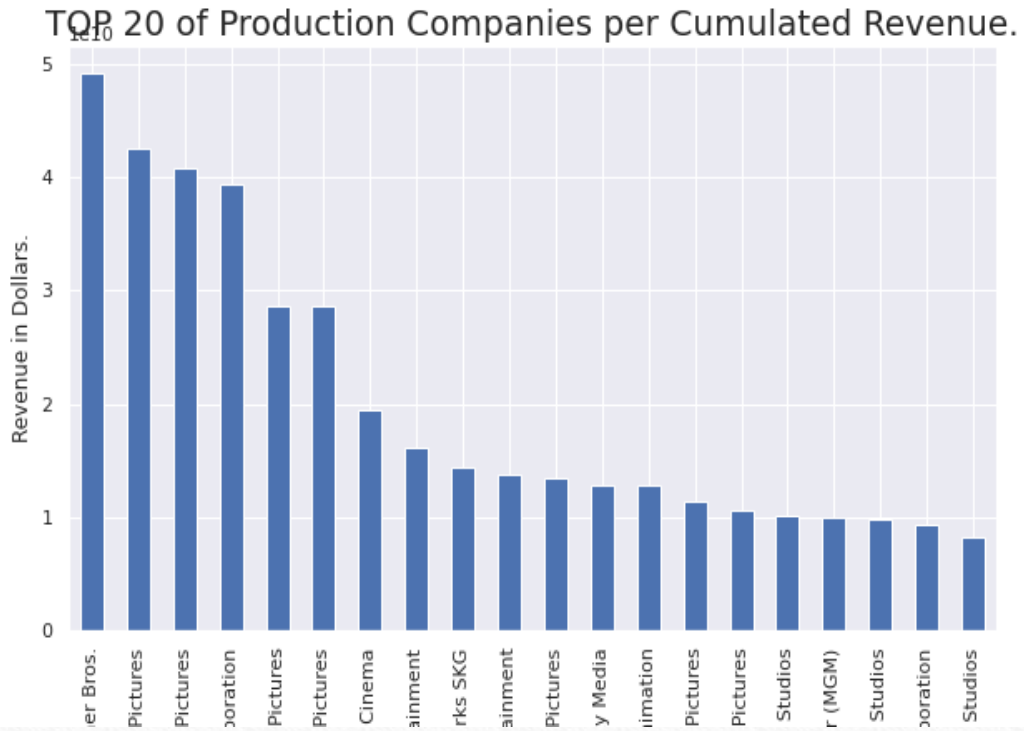
❖ Production Companies vs Number of Movies.



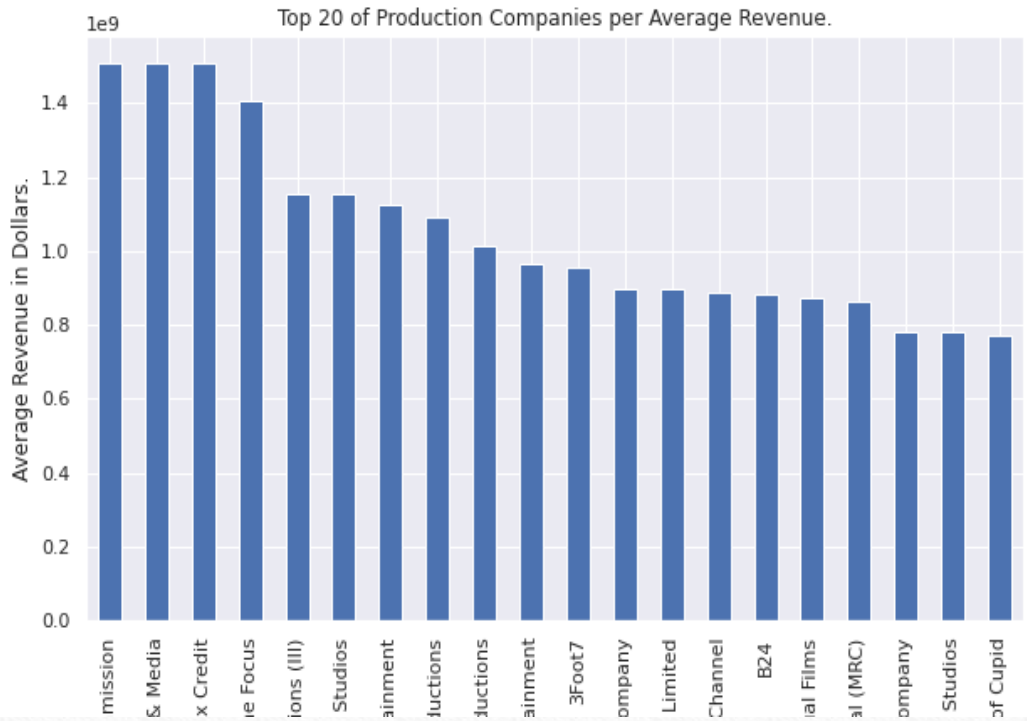
❖ Top 20 Production Companies.



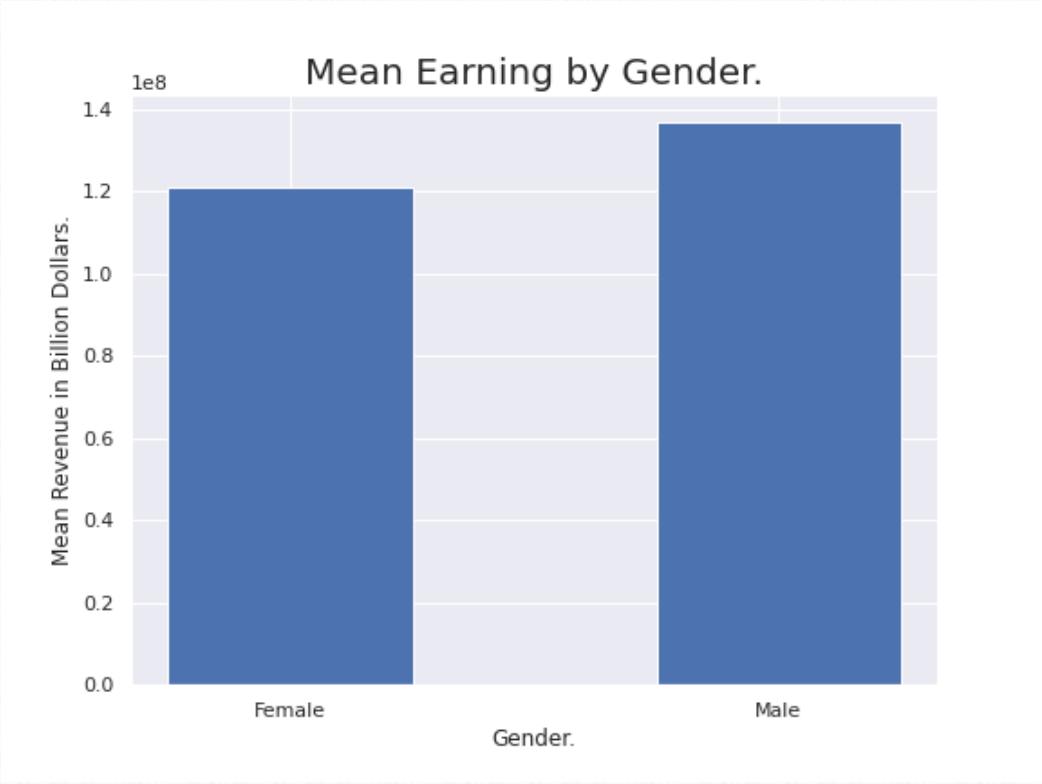
❖ Cumulative Revenues of Top 20 Production Companies.



❖ Average Revenues of Top 20 Production Companies.



❖ Gender vs Mean Revenue.



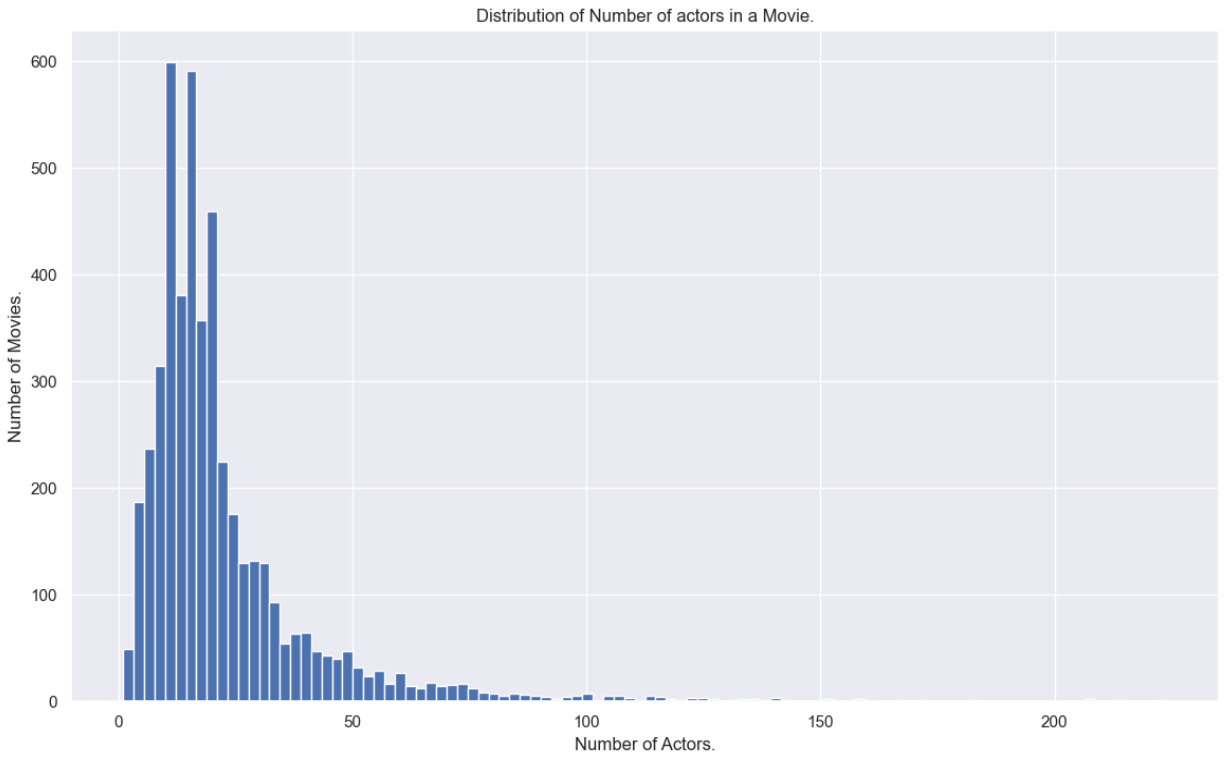
❖ Total actor of each Gender.

```
actor_gender
2    48288
0    33790
1    24167
Name: count, dtype: int64
```

❖ Total crew of each Gender.

```
crew_gender
0    74809
2    43000
1    11764
Name: count, dtype: int64
```

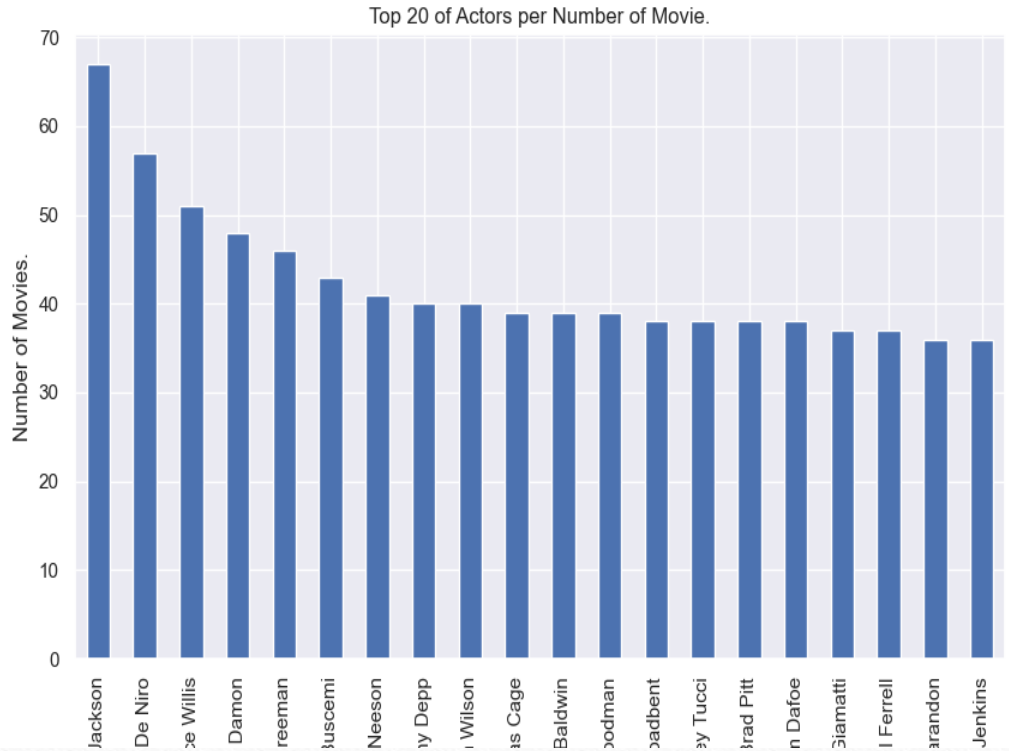
❖ Number of Movies vs Number of Actors.



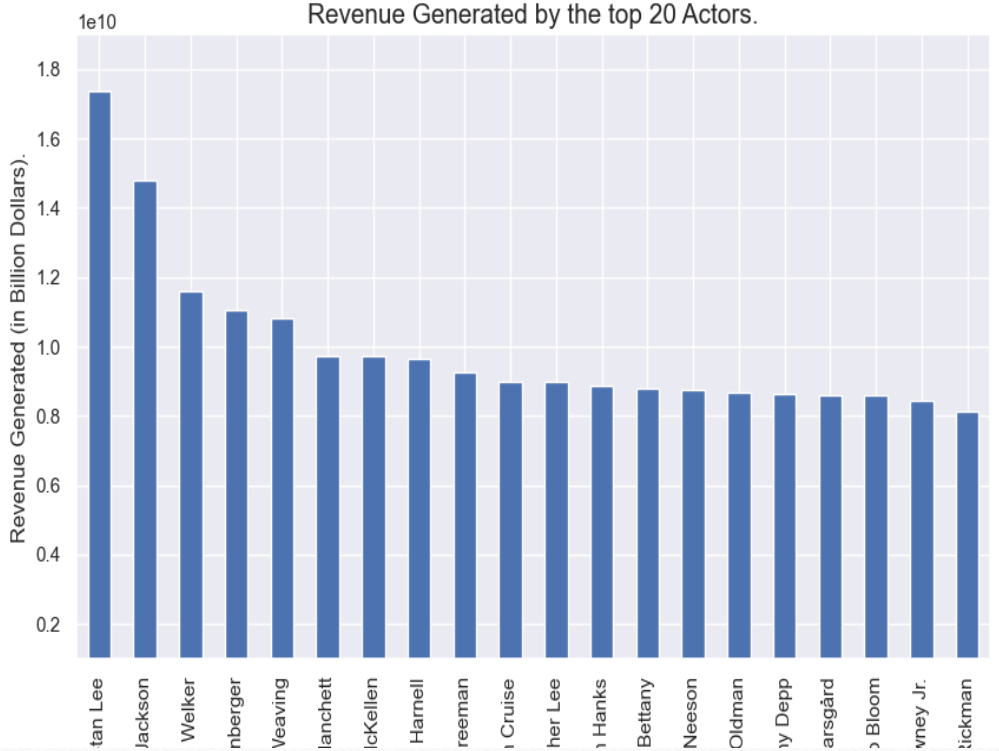
❖ Top 10 Movies with more No. of Actors.

title	Number of actors
Rock of Ages	224
Mr. Smith Goes to Washington	213
Jason Bourne	208
Les Misérables	208
You Don't Mess with the Zohan	183
Real Steel	172
Star Trek	168
Oz: The Great and Powerful	159
The Dark Knight Rises	158
Batman v Superman: Dawn of Justice	152

❖ Number of Movies vs Actors.



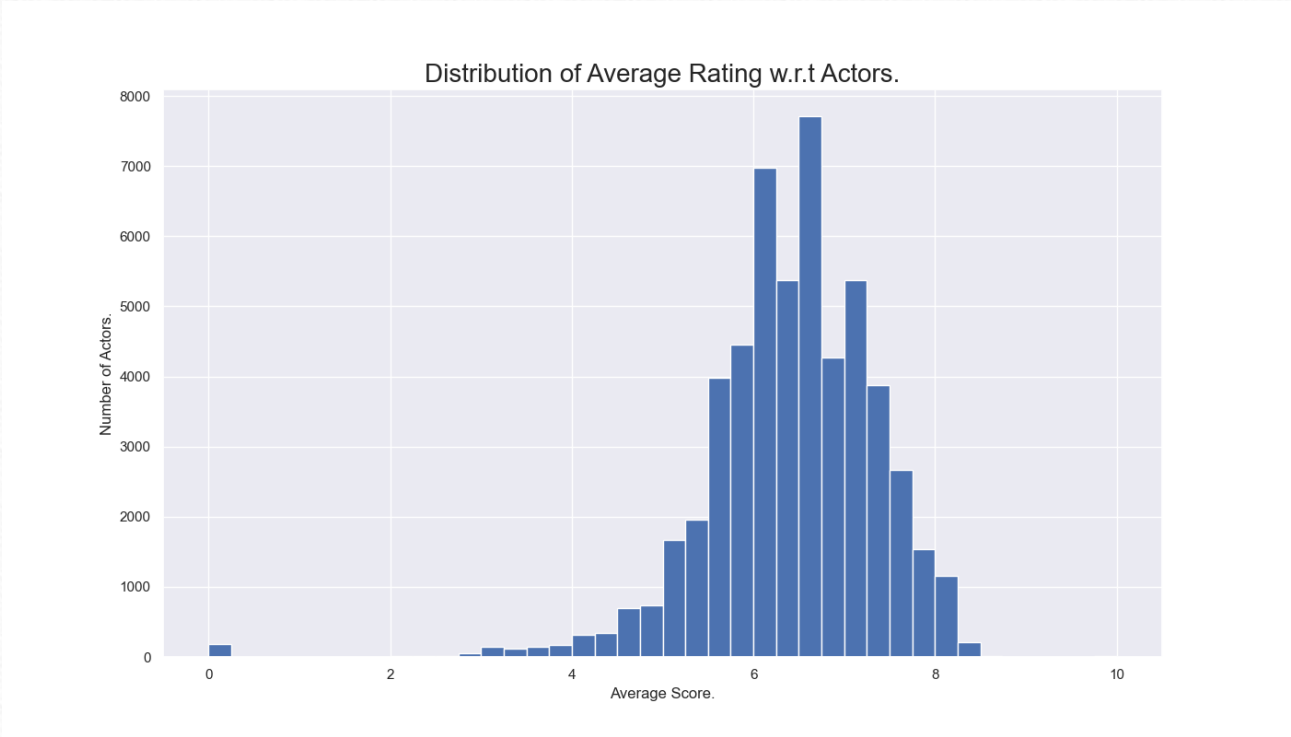
❖ Revenue Generated vs Actors.



❖ Actors with more Vote Count.

actor	vote_average
Patricia Wettig	10.0
Mel England	10.0
David Artus	10.0
Kevin Furlong	10.0
Travis Betz	10.0

❖ Average Score vs Actors.



- Crew Details.

```
title            4772
crew_name       52228
crew_job          418
crew_department    12
crew_gender         3
dtype: int64
```

- Total No. of people with respective Crew Job.

```
crew_job
Producer                    10205
Executive Producer           6177
Director                     5163
Screenplay                   5008
Editor                       4699
Casting                      4447
Director of Photography      3676
Art Direction                3338
Original Music Composer      3153
Production Design            2837
Name: count, dtype: int64
```

- Crew Departmets with No. of People working in it

```
crew_department
Production          27674
Sound               16174
Art                 14853
Crew                13826
Costume & Make-Up   11188
Writing             10686
Camera               9204
Directing            8146
Editing              7855
Visual Effects       7553
Lighting             2410
Actors                  4
Name: count, dtype: int64
```

❖ Top 5 Crew Members with more Movies.

crew_name	title
Robert Rodriguez	104
Steven Spielberg	84
Avy Kaufman	83
Mary Vernieu	82
Deborah Aquila	75

❖ Top 10 Crew with respective genre.

genres	crew_name	title
Action	Robert Rodriguez	76
Drama	Avy Kaufman	64
Action	Stan Lee	45
	Luc Besson	45
Comedy	Christophe Beck	43
	Woody Allen	43
Thriller	Robert Rodriguez	43
Family	Robert Rodriguez	42
Thriller	Deborah Aquila	41
Drama	Mary Vernieu	40

❖ Top 10 crew members with most crew job and title.

genres	crew_name	crew_job	title
Drama	Avy Kaufman	Casting	64
Thriller	Deborah Aquila	Casting	41
Comedy	Christophe Beck	Original Music Composer	41
Drama	Mary Vernieu	Casting	39
	Francine Maisler	Casting	39
Thriller	Tricia Wood	Casting	38
Drama	Deborah Aquila	Casting	37
Thriller	Mary Vernieu	Casting	36
Action	Dan O'Connell	Foley	36
	Joel Silver	Producer	33

❖ Top 10 genres with most particular crew jobs.

crew_job	genres	title
Producer	Drama	5077
	Comedy	3381
	Thriller	3130
	Action	2843
Executive Producer	Drama	2827
Director	Drama	2400
Screenplay	Drama	2174
Editor	Drama	2154
Casting	Drama	2056
Executive Producer	Thriller	2033

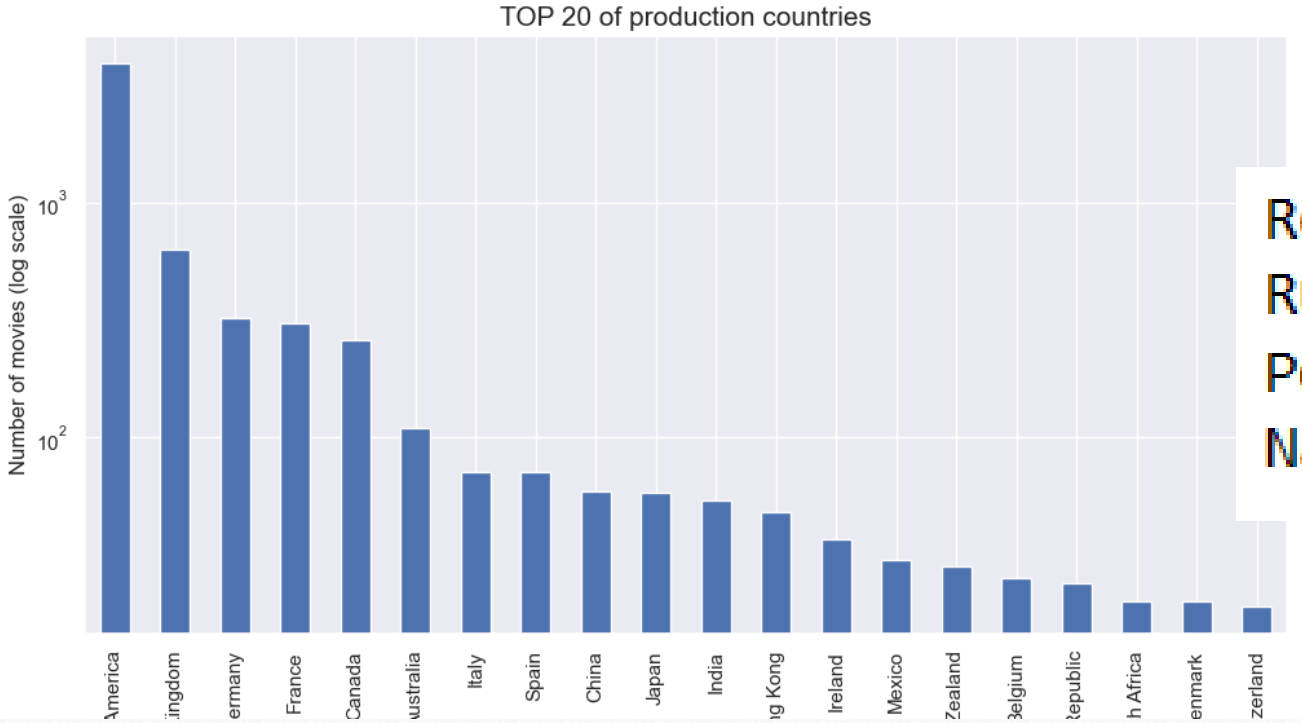
❖ Top 10 crew w.r.t Revenue.

crew_job	crew_name	revenue
Original Music Composer	Hans Zimmer	20928347941
Orchestrator	Kevin Kaska	18343111015
Original Music Composer	John Williams	17840602951
Compositors	Brian N. Bentley	17112706336
Executive Producer	Stan Lee	15566001300
Original Music Composer	James Newton Howard	15205930063
Casting	Sarah Finn	14343514479
Foley	Dan O'Connell	14018667060
Original Music Composer	Danny Elfman	12615749180
	John Powell	12404611924

❖ Top 10 Keywords.

keywords	id
woman director	324
independent film	318
duringcreditsstinger	307
based on novel	197
murder	189
aftercreditsstinger	170
violence	150
dystopia	139
sport	126
revenge	118

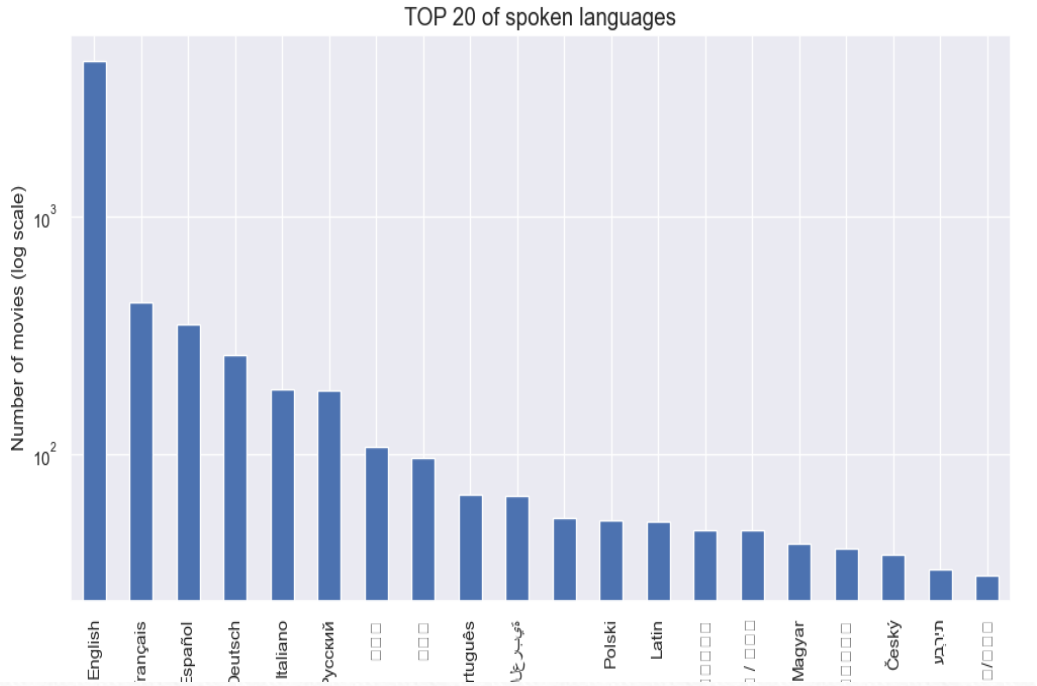
❖ Top 20 Production Countries.



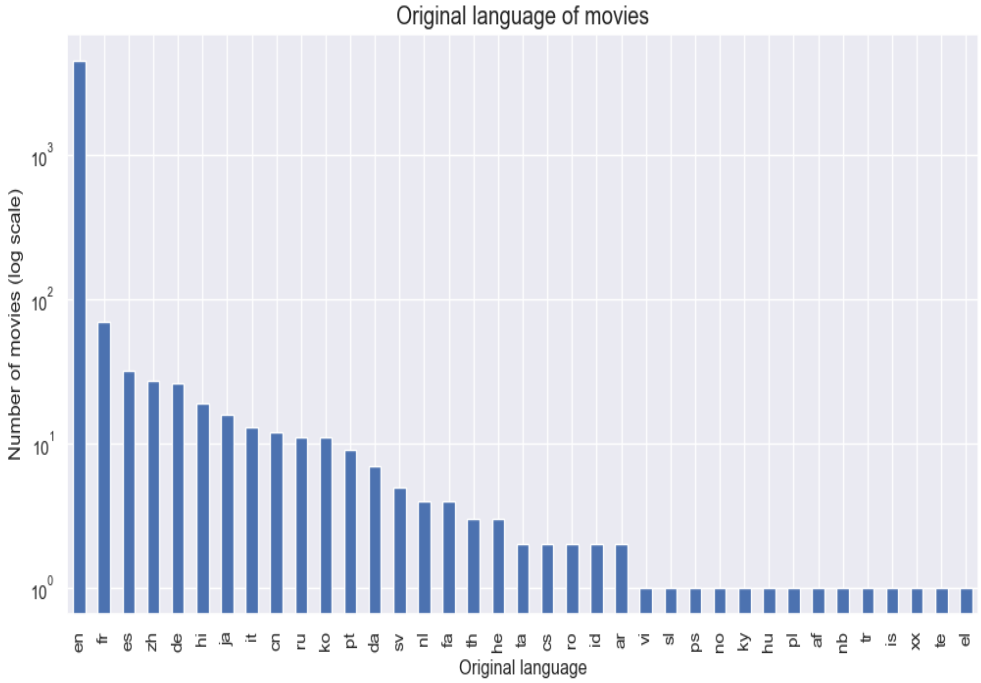
❖ Movies Status.

```
Released          4791
Rumored              5
Post Production      3
Name: status, dtype: int64
```

❖ Spoken Languages.



❖ Original Languages.



Hybrid Movie Recommender Model

- Methodology for Movie Selection and Filtering :
  1. Keyword, Tagline, and Genres Filtering.
  2. Vote Count Selection.
  3. Overview-Based Filtering.
- Data Preprocessing for Model :
  1. Import and loading of datasets into memory.
  2. Identified and removed all missing values from the dataset.
  3. Encompass essential attributes such as movie title, description, genres, keywords, tagline, user ratings, and vote counts.
  4. Extract the genres and keywords information.
  5. Merge them to prepare for the encoding process.

Hybrid Movie Recommender System

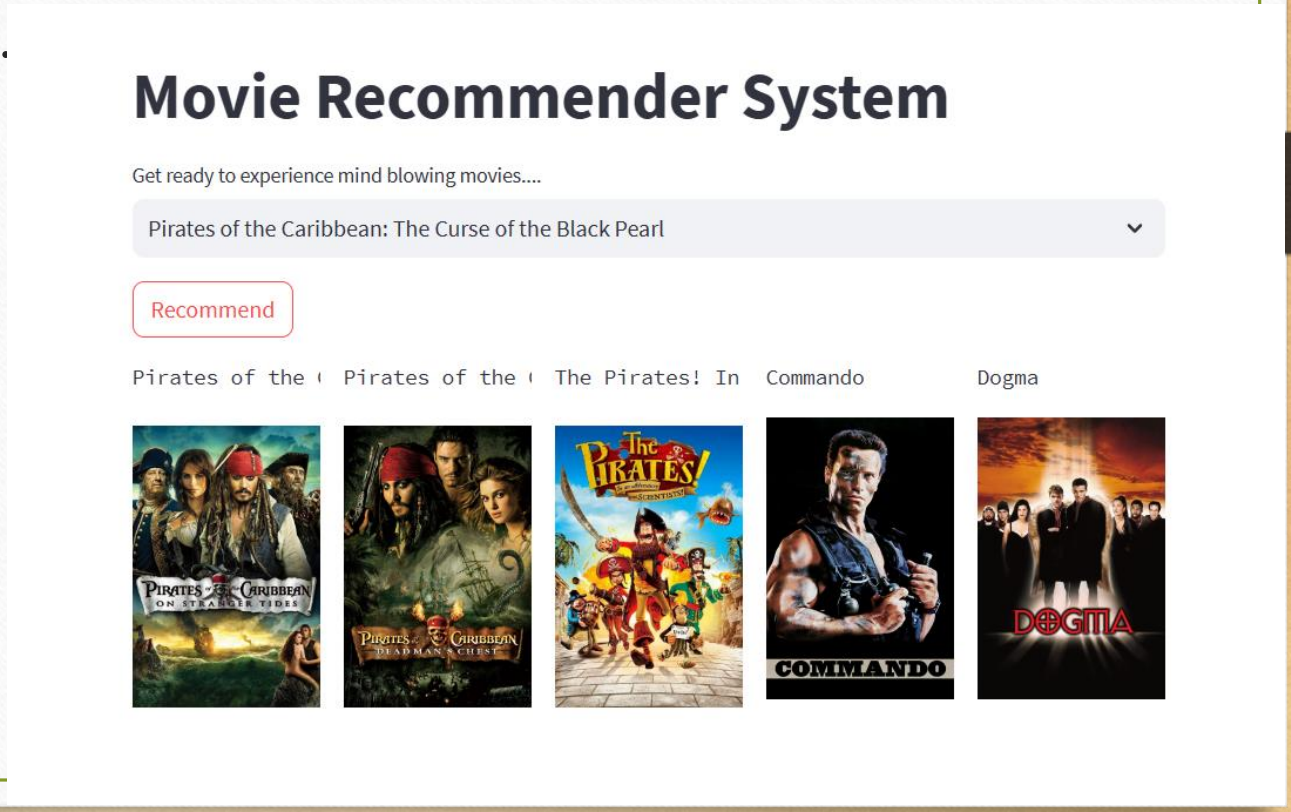
- Text Vectorization and Similarity Findings :
  1. In this case, use the CountVectorizer to encode the data. For this purpose, I will use the stop words provided by scikit-learn as the default, which excludes words like ’the’, ’and’, ’a’, ’an’, ’in’, ’of’, ’to’, etc.
  2. Cosine similarity : is a measure of similarity between two non-zero vectors of an inner product space that measures the cosine of the angle between them.
     - Ranges from -1 to 1, 1 - identical and 0 - dissimilar. Higher values of cosine similarity indicate greater similarity between the vectors.
     - cosine-similarity = (A . B) / (||A|| ||B||)
     - Retrieve the movie ID that the user just watched and find its similarity with other movies.

Function for Hybrid Recommender System

- Using the title of the movie, the recommender system will suggests top 5 similar movies that user has just watched.
- Obtain the movie index, using cosine-similarity matrix determine the 100 most similar movies.
- Select the top 50 movies from the 100 movies based on their vote counts.
  - Weighted Rating (WR) = (v / (v + m)) * R + (m / (v + m)) * C •
  - C = Mean vote (or rating) across all movies.
  - m = Minimum votes (or ratings) required for the movie to be considered.
  - Function WR = Calculate the weighted average ratings using the provided formula.
- Sort the values and pick the highest scores to select the top 50 movies for recommendation.
- Filter the movies based on their content similarity, and for this, we'll be using the movie overview.
- We have obtained the final output of our basic movie recommendation system!, I am finding the movies sequentially based on the previous results.

Hybrid Movie Recommender System Implementation

1. Data Retrieval and Preprocessing.
2. User Interaction.
3. Recommendation Function.
4. Movie Similarity and Filtering.
5. Content-Based Filtering.
6. TF-IDF Vectorization.
7. Recommendation Output.
8. Streamlit Interface.



Summary and Conclusion

- **Information Overload** :
  - Provides users with tailored movie recommendations.
  - Navigate and discover those movies that truly match their preferences.
- **Diverse User Preferences** :
  - By blending multiple recommendation techniques, it adapts to users' changing tastes.
- **Cold Start Problem** :
  - New users, with limited interaction history, receive meaningful recommendations based on content-based attributes.
- **Limited Exploration** :
  - Offers diverse movie suggestions, enhancing user engagement and satisfaction.

Way Forward

- The lessons learned from this project may serve as a stepping stone for future enhancements :
  1. Real-world deployment.
  2. The inclusion of user feedback mechanisms for continuous system improvement.
  3. Online Machine Learning.

Any Questions ?

Thank You !